

객체지향시스템설계 Term Project #1

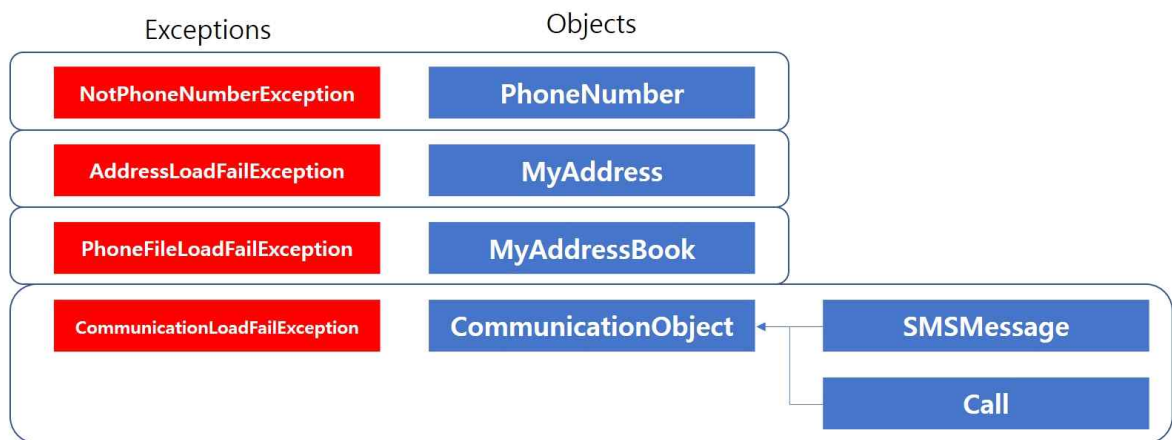
<2013011769 강승원, 이호준>

□ 팀 구성과 역할

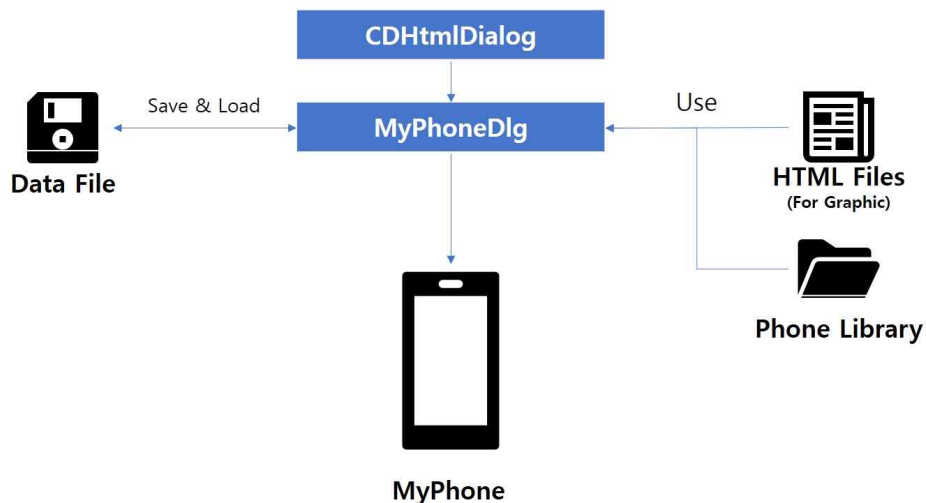
Work	Created By	Helped By	Used Program
프로그램 설계	이호준, 강승원		
Phone Library	이호준	강승원	Visual Studio 2017
Test Application	강승원	주변 친구들(Tester)	Visual Studio 2017 Illustrator (Design)

□ 구성도

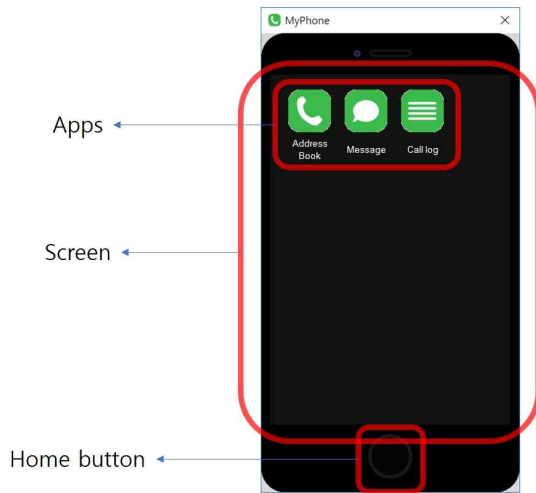
1) Phone Library 클래스 구성도



2) Test Application 구성도



□ 사용자 인터페이스(User Interface)



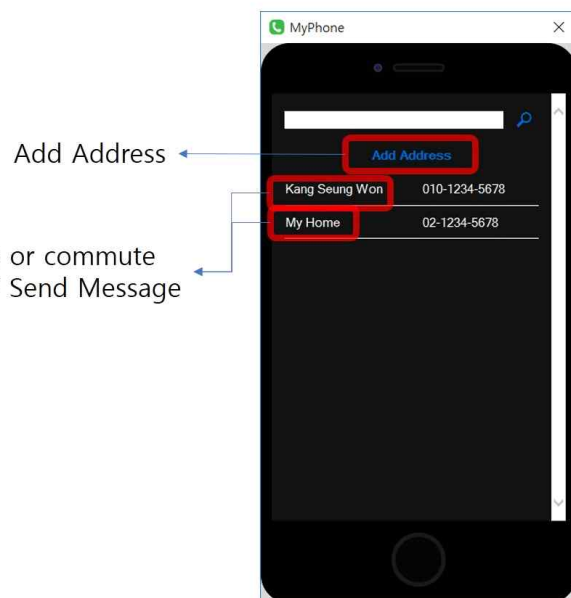
프로그램을 설계할 때 Phone Library 와 Test Application을 구현 하는 것이므로 최대한 GUI를 스마트폰(아이폰)과 비슷하게 구현을 했다. I-Phone 7 매트블랙을 모델로 했으며 위 그림은 메인 화면으로 사용법은 아이폰과 거의 동일하다.

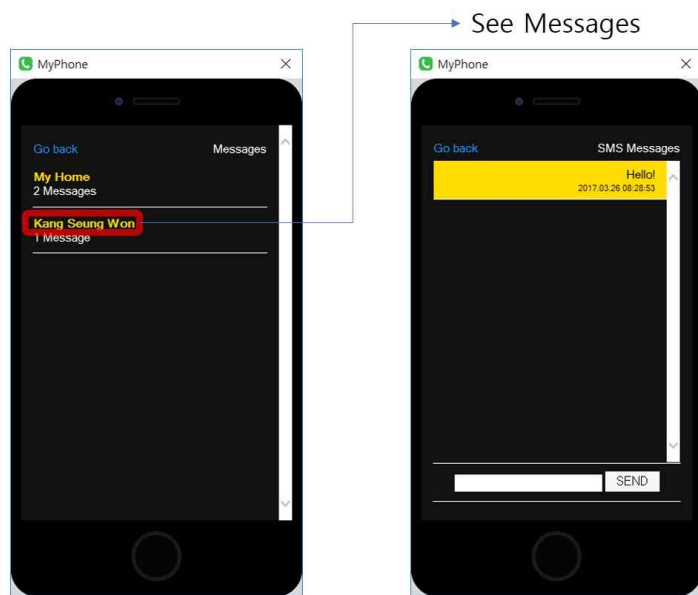
기능	설명
Apps	스마트폰(아이폰)에서 앱과 같은 역할. 현재 3개의 어플리케이션이 있으며, 각각 AddressBook, Message, Call log(전화 기록) 이다.
Screen	터치스크린. 어플리케이션 화면이 이 Screen에서 나타난다.
Home Button	메인 화면으로 가는 버튼이다.

AddressBook App

AddressBook 앱은 주소 추가/삭제/수정 등 주소록 변경하는 것부터 등록된 주소로 전화(Call) 나 메시지(Sending Message)를 보낼 수 있는 앱이다. “Add Address” 버튼을 눌러서 주소를 추가할 수 있으며, 그 외(삭제, 수정, 전화, 메시지 보내기)의 기능은 목록에 있는 이름을 누르면 사용이 가능하다.

Address Info or commute
Modify / Call / Send Message





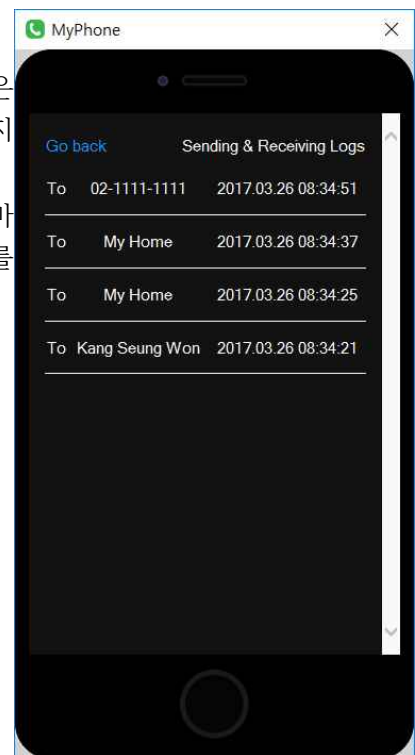
Message App

Message App은 현재 내가 보내거나 나에게 온 메시지의 목록을 보여주며, 목록은 최신 순으로 Sorting 한 상태로 보여준다. 해당 이름을 누르면 메시지를 볼 수 있다.

Call log App

Call Log 앱은 전화 송수신 기록을 보여주는 앱이다. 첫 열에는 송신은 To 로 수신은 From 으로 뜨고, 두 번째 열은 이름(주소록에 등록되지 않은 주소면 전화번호가 뜬다), 세 번째 열은 시간이 뜬다.

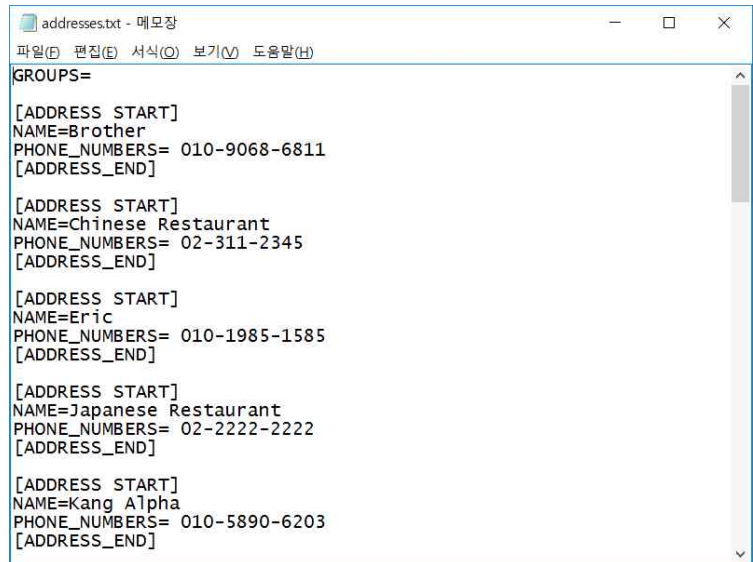
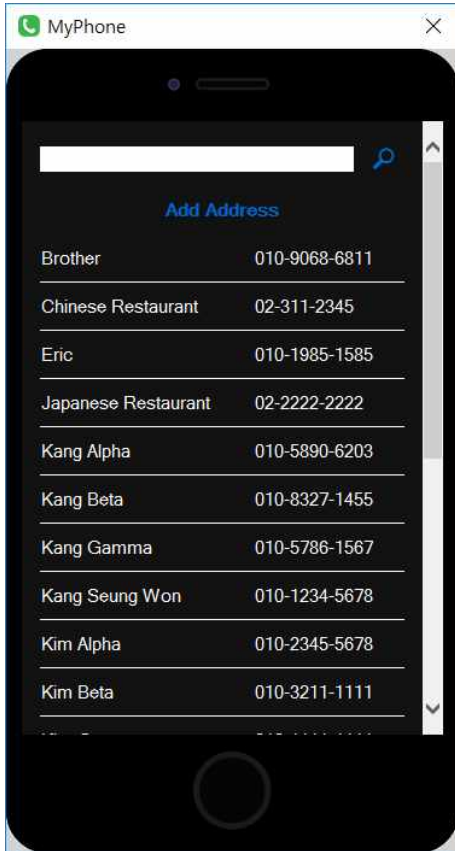
송신은 AddressBook 앱에서 전화를 걸어서 가능하지만 수신은 내 스마트폰에서 나에게 전화를 걸 수 없으니 기능이 제공되지 않고, 텍스트를 수정해야만 가능하다.



□ Test 과정(과제 PPT 6번)

① 20 개의 Entry를 생성하여서 Phone Library의 In-Memory Collection 과 File에 저장하십시오. 이를 File에서 읽어, 깔끔하게 화면에 Display하세요.

전화번호 20개를 추가하였으며, 화면에 전화목록 20개와 텍스트 파일에 20개가 추가 된 것을 확인할 수 있었다.

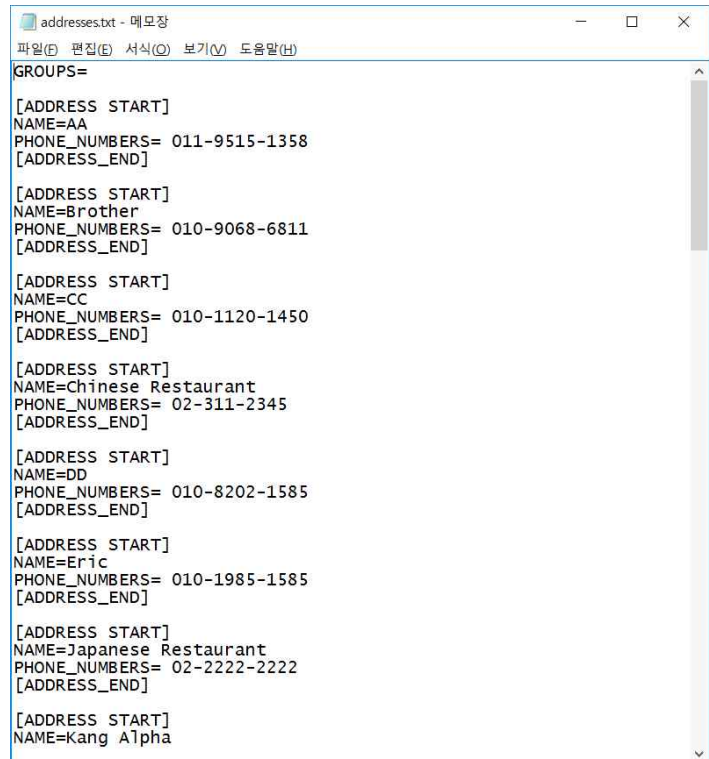
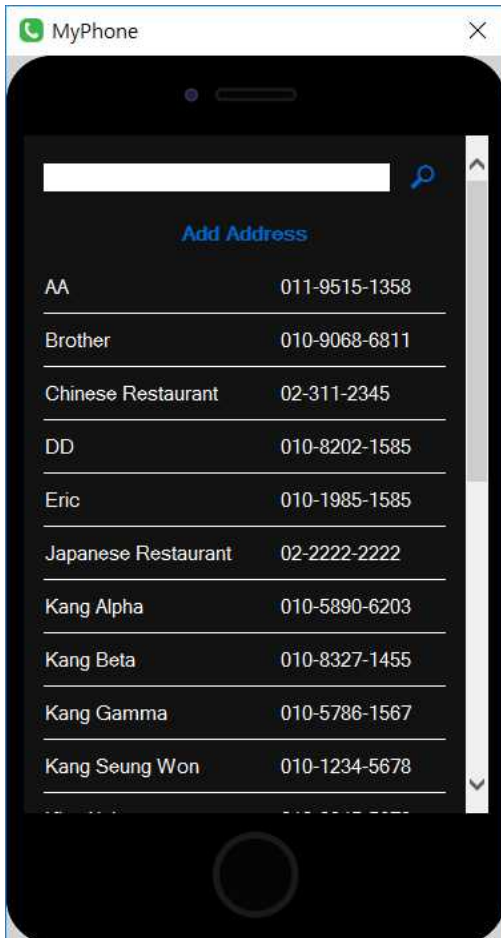


② 이름의 Alphabet 순서로 Sorting하세요. 화면에 Display하시고 File에 Save하세요

전화번호를 추가할 때마다 Sorting 한 상태로 Display를 하며, 파일을 Save 할 때도 Sorting 된 상태로 보여준다.

③ Test Application에서 전화번호부의 10 개의 Entry를 추가로 생성하여서 추가하세요. 이때 Sorting 한 상태를 계속 유지해야 합니다. 이를 확인할 수 있도록 보여주세요. 그리고 이를 File에 저장하고 읽으세요. 여전히 Sort된 상태에 있나요?

AA와 DD를 추가했을 때 아래 그림과 같이 Sorting 한 상태를 계속 유지된 상태로 추가가 된다. 또한 파일도 Sorting 된 상태와 동일하게 나타난다.



④ Text Files을 Editor로 편집하여서 3개의Entries를 추가하고, 2개를 변경하고, 1개를 지우세요. 이를 제대로 읽어서 다시 Sorting하여서 보여주시고 저장하세요.

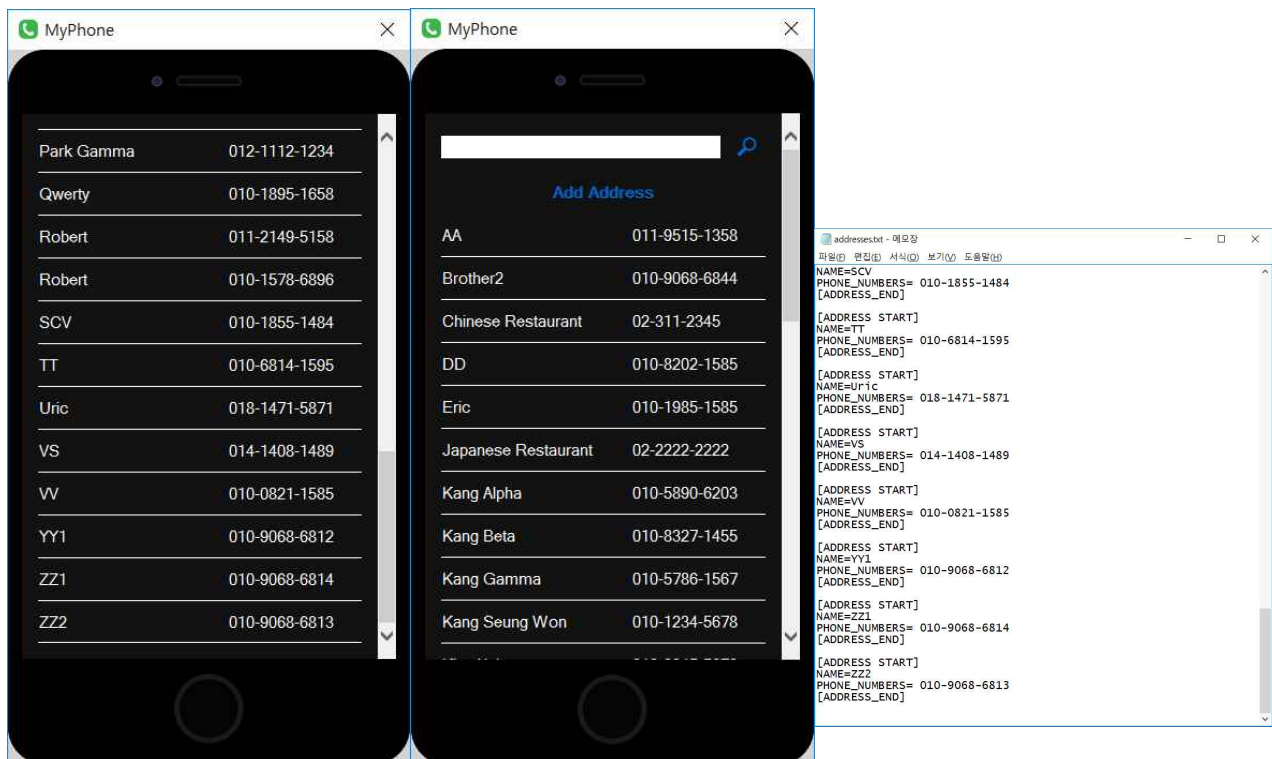
CC를 삭제하고, YY1, ZZ2, ZZ1를 추가했으며, Brother를 이름을 Brother2 로 전화번호를 010-9068-6811에서 010-9068-6844 로 바꿨다. 그림과 같이 프로그램을 실행을 하면, 로딩할 때 알아서 Sorting을 하고, 제대로 정렬된 상태로 보여지는 것을 확인할 수 있다. 또 저장을 하면 Sorting 된 상태로 저장이 됨을 확인할 수 있다.

```

addresses.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
GROUPS=
[ADDRESS START]
NAME=AA
PHONE_NUMBERS= 011-9515-1358
[ADDRESS_END]
[ADDRESS START]
NAME=Brother
PHONE_NUMBERS= 010-9068-6811
[ADDRESS_END]
[ADDRESS START]
NAME=CC
PHONE_NUMBERS= 010-1120-1450
[ADDRESS_END]
[ADDRESS START]
NAME=Chinese Restaurant
PHONE_NUMBERS= 02-311-2345
[ADDRESS_END]
[ADDRESS START]
NAME=DD
PHONE_NUMBERS= 010-8202-1585
[ADDRESS_END]
[ADDRESS START]
NAME=Eric
PHONE_NUMBERS= 010-1985-1585
[ADDRESS_END]
[ADDRESS START]
NAME=Japanese Restaurant
PHONE_NUMBERS= 02-2222-2222
[ADDRESS_END]
[ADDRESS START]
NAME=Kang Alpha

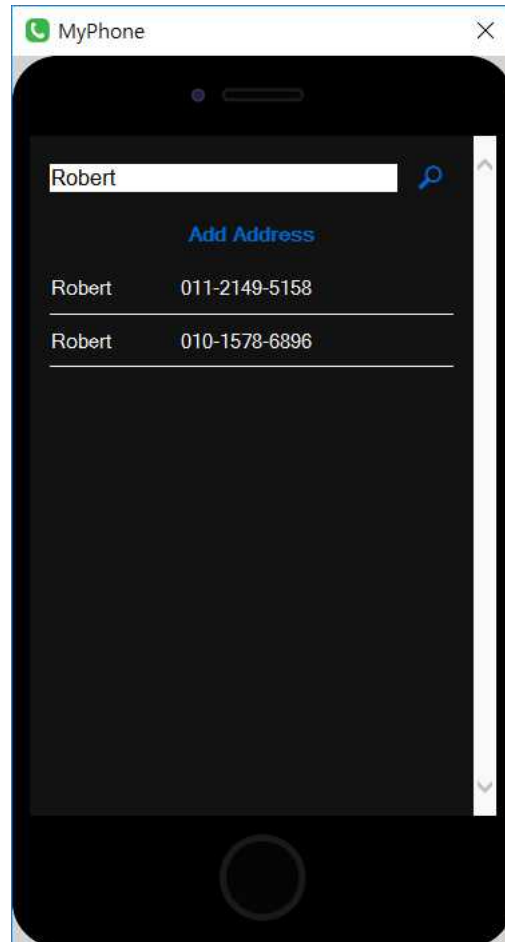
addresses.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
GROUPS=
[ADDRESS START]
NAME=YY1
PHONE_NUMBERS= 010-9068-6812
[ADDRESS_END]
[ADDRESS START]
NAME=ZZ2
PHONE_NUMBERS= 010-9068-6813
[ADDRESS_END]
[ADDRESS START]
NAME=ZZ1
PHONE_NUMBERS= 010-9068-6814
[ADDRESS_END]
[ADDRESS START]
NAME=AA
PHONE_NUMBERS= 011-9515-1358
[ADDRESS_END]
[ADDRESS START]
NAME=Brother2
PHONE_NUMBERS= 010-9068-6844
[ADDRESS_END]
[ADDRESS START]
NAME=Chinese Restaurant
PHONE_NUMBERS= 02-311-2345
[ADDRESS_END]
[ADDRESS START]
NAME=DD
PHONE_NUMBERS= 010-8202-1585
[ADDRESS_END]
[ADDRESS START]
NAME=Eric

```



⑤ 전화번호부에서 주어진 이름을 갖는 Entry를 찾는 기능을 구현하시고 Test하세요. 이를 결과로서 보여주세요.

Robert 라는 이름으로 검색을 했더니 이름이 Robert인 사람 2명이 나왔다. (Test 과정에서 Robert를 2명 생성했었음. 전화번호 중복은 허용이 되지 않게 했지만 이름 중복은 허용이 되게 만들었음.)



⑥ 전화번호를 가지고 Entry를 찾는 효율적인 Algorithm에 대한 설계를 하고, 구현 및 Test를 하십시오.

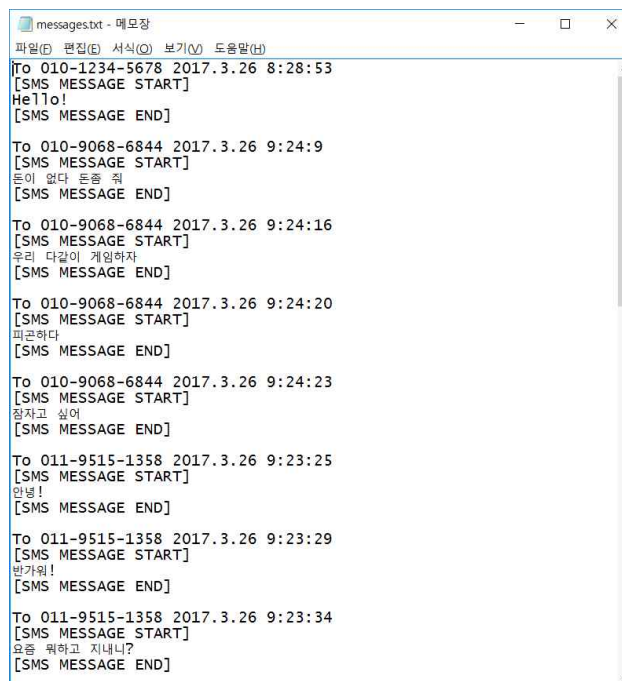
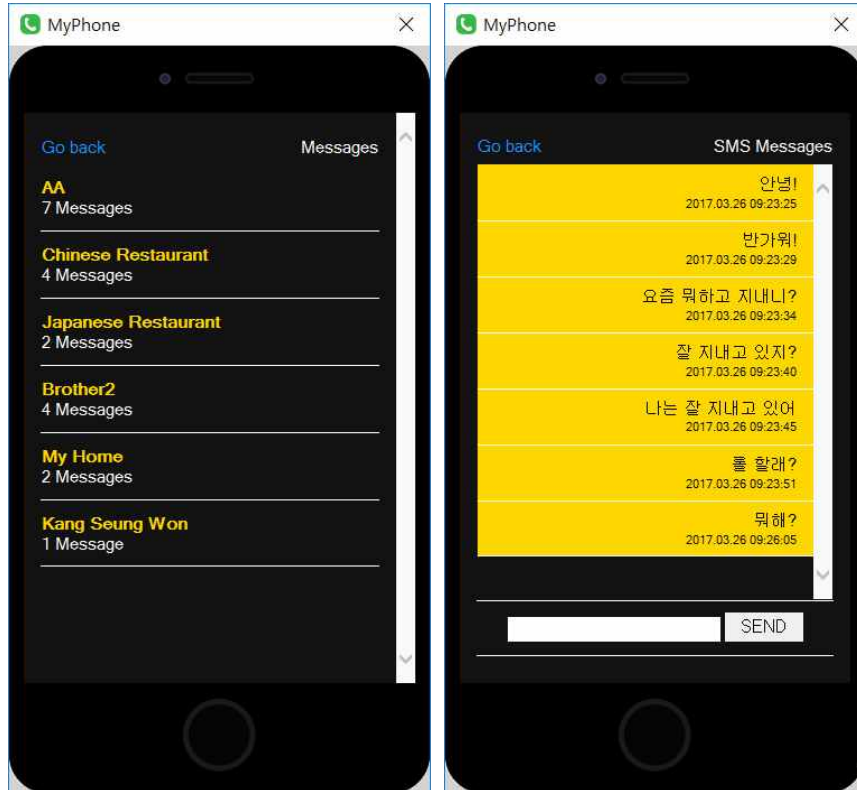
MyAddressBook 클래스는 단순히 Address를 저장하는 자료구조(m_myAddresses) 뿐만 아니라 주소록에 있는 모든 전화번호와 그 전화번호를 갖는 이름을 저장하는 자료구조(m_namesByNumbers 멤버변수, 일종의 해쉬테이블)를 가지고 있으며, SearchAddress(PhoneNumber& number) 메소드를 호출해서 MyAddress 배열의 Entry 값을 얻을 수 있다.

전화번호로 해당 Address의 index를 찾는 함수는 처음에 전화번호로 이름값을 얻는데 걸리는 시간 $O(\log n)$, 전화번호로 얻은 이름으로 Address의 인덱스를 찾는데 걸리는 시간 $O(\log n)$ 으로, Binary Search를 2번 수행하여 총 걸리는 시간은 $O(\log n)$ 이다. (단, n 개의 Address가 있고, Address의 Name이 모두 동일하면 걸리는 시간은 $O(n)$ 이지만, Address 개수 n 이 Time Complexity에 영향을 줄만큼 Address 개수가 많고, 모든 Address의 이름이 같은 경우는 굉장히 드물기 때문에(거의 없다고 봐도 무관) 크게 문제되지 않는다고 생각한다)

Message나 Call log에서 번호가 없을 때는 그냥 전화번호로 뜨고, 번호가 있을 때는 이름으로 뜨게 만들었는데 여기서 이 기능을 사용하고 있으며, (이 알고리즘을 사용하지 않아도 번호로 이름을 찾을 수 있지만 속도가 더 느리다) 이름이 잘 뜨는 것을 확인할 수 있다.

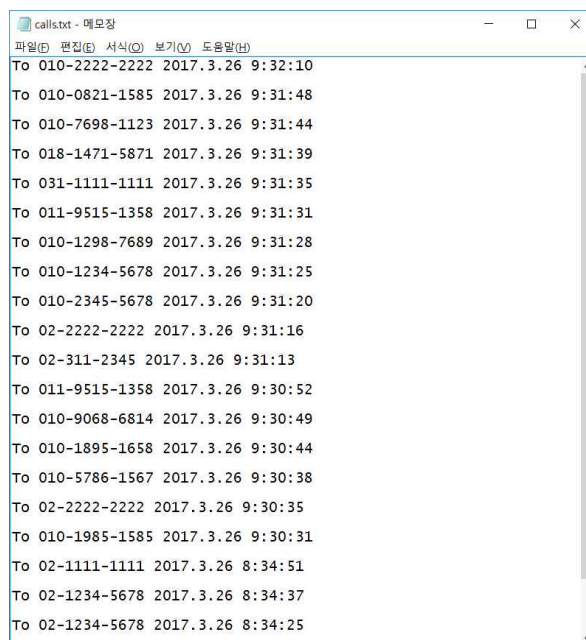
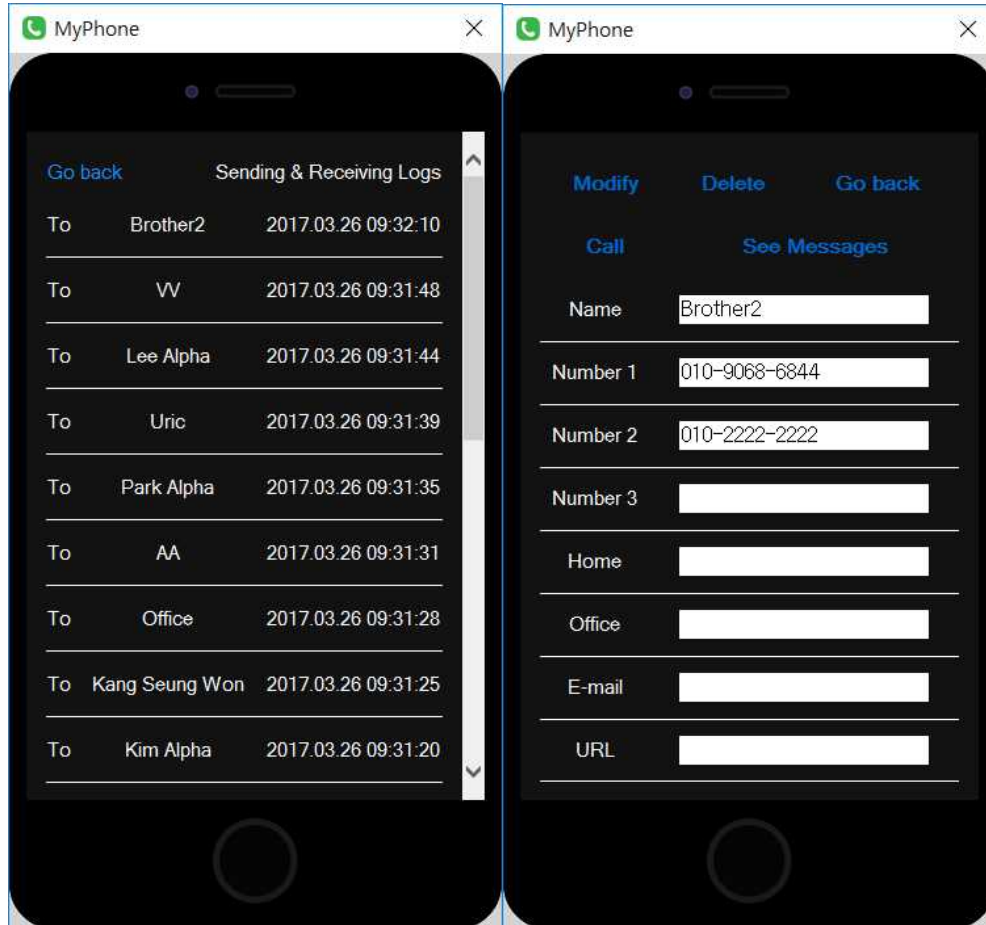
⑦ 이를 바탕으로 수발신 SMS Message를 Browse하고, 선택해서 보여주는 기능을 구현하세요. SMS 메시지를 20개 만들어서 Test하여주세요. 이를 File에 Save하고 읽어서 보여주세요.

아래와 같이 Message App을 들어가면, Message 목록을 볼 수 있다. 위에 있을수록 최신 메시지이며, 이름(노란색 글씨)를 클릭하면, 메시지 내용을 볼 수 있다. 파일에도 저장된 메시지 내용을 볼 수 있다.



⑧ 수신전화에 대해서 Caller ID에 기반하여, 필요시 발신자의 정보를 보여주는 기능을 구현하세요. 20개의 수발신전화 Entry를 만드시고 File에 Save하고 읽어서 보여주세요.

Call log 앱을 통해 송수신 목록을 볼 수 있으며, 이름이나 전화번호를 클릭하면 해당 발신자의 정보를 볼 수 있다. Save 된 파일을 열어보면 시간에 정렬된 상태로 수발신 기록이 나와있음을 볼 수 있다.



□ 결론(Conclusion)

강승원 : 처음에 설계를 할 때 나름 필요한 기능을 모두 넣었다고 생각했는데 만들다보니 빠진 기능이 몇몇 있고, 제작 과정에서 설계를 좀 다시하고 또 교수님이 말한 대로 Exception 등 기능을 추가하는데도 기존 코드를 수정해서 약간 꼬인 부분이 없잖아 있었던 것 같다. 또 코드가 길어지니깐 원래 습관대로 주석 처리를 하지 않고 막 코딩하는 습관이 나와 버리기도 했다. (Unreadability) 하지만 완성이 되었을 때 나름 뿌듯했고, Exception 의 중요성, 설계의 중요성을 다시 깨달을 수 있었다. 또 프로그램을 만들면서 좋지 않은 습관을 고쳐야겠다는 생각이 들었다. 그리고 라이브러리와 테스트 프로그램을 분리함으로써 실제 프로그래머들이 어떤 식으로 코딩을 하는 지 잘 알 수 있는 계기가 되었던 것 같다.

이호준 : 객체지향적 설계를 실습함으로써 객체지향의 장점 및 특징들을 배울 수 있었다. 첫째, 번호의 주인, 전화번호 등의 정보들로 이루어진 하나의 구조적인 자료(my address)를 객체로 개념화함으로써 Abstraction 과정을 직접 경험할 수 있었다. 둘째, 테스트 파일을 따로 만듦으로써 Encapsulation을 통한 작업의 효율화를 직접 경험할 수 있었다. 셋째, 예외처리를 설계함으로써 이 부분을 연습해볼 수 있었다.

팀 프로젝트를 통해 배운 점도 있었지만 동시에 부족한 점도 많이 느낄 수 있었다. 먼저, 내가 얼마나 프로그래밍에 있어 경험이 부족했는지 알 수 있었다. 활용도가 높은 함수나 클래스들의 사용법을 미리 숙지해 놓은 상태였다면 훨씬 빠르고 효율적으로 프로젝트를 진행했을 터인데 그러지 못하여 매번 검색을 통해 활용법을 찾아야 했었다. 기본적으로 프로그래밍 경험을 더 늘리고 새로 알게 된 유용한 정보들을 따로 정리해두는 습관을 길러 부족한 점을 메꿔야 할 필요성을 느꼈다. 둘째, 깃허브 사용을 습관화하지 못하고 또 그것을 적극적으로 활용하지 못하여 다양한 버전으로 관리 하지 못했다. 사용법을 다시 한번 숙지하고 자주 사용하여 깃허브 환경에 익숙해 지고 스스로 다양한 프로그래밍을 깃허브에서 접해 보는 자세를 통해 부족한 점을 메꿔야겠다고 느꼈다.

사실 이번 프로젝트를 진행하면서 파트너이신 강승원 학우에게서 정말 많은 도움을 받았고 또 많은 점을 배울 수 있었다. 조안을 통해 디버깅 과정에서 사소한 도움을 받을 수 있었을 뿐만 아니라, vector나 iterator와 같은, 전에 알지 못했던 유용한 라이브러리 및 함수들을 사용해볼 수도 있었다. 부족한 파트너임에도 내색없이 프로젝트를 진행하고 많은 도움말을 주어 진심으로 감사함을 느낀다.